I claim:

1. A deadlock avoidance apparatus in a microprocessor having a speculative branch target address cache (BTAC), the apparatus comprising:

a first signal, for indicating a miss of a fetch address in an instruction cache, wherein said fetch address is a speculative branch instruction target address provided by the BTAC;

a second signal, for indicating said branch instruction wraps across two cache lines; and

a third signal, for indicating the BTAC predicted said branch instruction does not wrap across said two cache lines, whereby a second of said two cache lines is not fetched because the BTAC predicted said branch instruction does not wrap across said two cache lines; and

control logic, coupled to receive said first, second, and third signals, for invalidating said target address in the BTAC, in response to a true indication on said first, second, and third signals.

2. The apparatus of claim 1, wherein said control logic is further configured to cause the microprocessor to branch to said branch instruction after said invalidating said target address in the BTAC.

3. The apparatus of claim 1, wherein an instruction formatter determines said branch instruction wraps across said two cache lines by decoding a first of said two cache lines.

4. The apparatus of claim 3, wherein said instruction cache provides said first of said two cache lines.

5. The apparatus of claim 1, wherein said branch instruction target address is speculative because said target address is only a prediction and is not guaranteed to be a correct target address of said branch instruction.

6. The apparatus of claim 1, wherein said branch instruction target address is speculative because said target address may be a target address of a different branch instruction.

7. The apparatus of claim 1, wherein said branch instruction target address is speculative because said branch instruction may not be present in said two cache lines.

8. The apparatus of claim 1, wherein said second of said two cache lines is not fetched comprises said second of said two cache lines is not fetched from a memory coupled to the microprocessor.

9. A pipelined microprocessor for avoiding a deadlock condition, comprising:

a branch target address cache (BTAC), for providing a speculative target address of a branch instruction in response to an instruction cache fetch address; and

control logic, coupled to said BTAC, for invalidating said speculative target address in said BTAC in response to detecting a condition in which:

said speculative target address misses in said instruction cache after said instruction cache provides a first cache line in response to said fetch address, wherein said first cache line contains only a first portion of said branch instruction; and

said BTAC incorrectly predicts that said branch instruction is wholly contained within said first cache line, thereby causing a second cache line containing a second portion of said branch instruction not to be fetched from said instruction cache.

10. The microprocessor of claim 9, wherein said control logic is further configured to cause the microprocessor to branch back to said fetch address after said invalidating said speculative target address in said BTAC in response to detecting said condition.

11. The microprocessor of claim 9, further comprising:

an instruction fetcher, coupled to said control logic, for fetching missing cache lines into said instruction cache from a memory coupled to the microprocessor, wherein said instruction fetcher is configured not to fetch missing cache lines from speculative target addresses provided by said BTAC.

12. The microprocessor of claim 9, further comprising:

an instruction formatter, coupled to said control logic, for decoding said first cache line and generating a signal to said control logic indicating said branch instruction is not wholly contained within said first cache line.

13. The microprocessor of claim 12, further comprising:

an instruction buffer, coupled to said instruction cache, for receiving said first cache line from said instruction cache and storing said first cache line while said instruction formatter decodes said first cache line.

14. The microprocessor of claim 12, further comprising:

execution logic, coupled to said control logic, for detecting and correcting BTAC mispredictions, wherein said execution logic does not detect said BTAC incorrectly predicts that said branch instruction is wholly contained within said first cache line because said instruction formatter stalls waiting for said second cache line to be fetched.

15. The microprocessor of claim 14, wherein said execution logic resides in a stage of the microprocessor pipeline below said instruction formatter.

16. The microprocessor of claim 9, wherein an instruction set of the microprocessor comprises instructions of variable length.

17. The microprocessor of claim 9, wherein said instruction set conforms substantially to an x86 architecture instruction set.

18. A method for avoiding a deadlock condition in a microprocessor having an instruction cache and a speculative branch target address cache (BTAC), the method comprising:

generating a speculative target address of a branch instruction partially contained in a first cache line provided by the instruction cache in response to a first fetch address, in response to applying the first fetch address to the BTAC;

providing said target address as a second fetch address to the instruction cache without fetching a next cache line sequential to the first cache line, in response to the BTAC predicting the branch instruction is wholly contained in the first cache line;

determining the BTAC incorrectly predicted the branch instruction is wholly contained in the first cache line;

detecting a miss of the target address in the instruction cache; and

invalidating the target address in the BTAC, in response to said determining and said detecting.

19. The method of claim 18, further comprising:

branching the microprocessor to an address of the branch instruction, after said invalidating.

20. The method of claim 19, wherein said branching the microprocessor to an address of the branch instruction comprises providing the first fetch address to the instruction cache as a next fetch address.

21. The method of claim 19, wherein said branching the microprocessor to an address of the branch instruction comprises assigning the address of the branch instruction to an instruction pointer register of the microprocessor.

22. A computer data signal embodied in a transmission medium, comprising:

computer-readable program code for providing a microprocessor, said program code comprising:

first program code for providing a branch target address cache (BTAC), for providing a speculative target address of a branch instruction in response to an instruction cache fetch address; and

second program code for providing control logic, coupled to said BTAC, for invalidating said speculative target address in said BTAC in response to detecting a condition in which:

said speculative target address misses in said instruction cache after said instruction cache provides a first cache line in response to said fetch address, wherein said first cache line contains only a first portion of said branch instruction; and

said BTAC incorrectly predicts that said branch instruction is wholly contained within said first cache line, thereby causing a second cache line containing a second portion of said branch instruction not to be fetched from said instruction cache.

23. A computer data signal embodied in a transmission medium, comprising:

computer-readable program code for providing a deadlock avoidance apparatus in a microprocessor having a speculative branch target address cache (BTAC), said program code comprising:

first program code for providing a first signal, for indicating a miss of a fetch address in an instruction cache, wherein said fetch address is a speculative branch instruction target address provided by the BTAC;

second program code for providing a second signal, for indicating said branch instruction wraps across two cache lines;

third program code for providing a third signal, for indicating the BTAC predicted said branch instruction does not wrap across said two cache lines, whereby a second of said two cache lines is not fetched because the BTAC predicted said branch instruction does not wrap across said two cache lines; and

fourth program code for providing control logic, coupled to receive said first, second, and third signals, for invalidating said target address in the BTAC, in response to a true indication on said first, second, and third signals.